Audio Player

AI1110: Probability and Random Variables
Indian Institute of Techonology Hyderabad

K S Ananth
CS22BTECH11029

1 Introduction

This report presents an analysis of a Python script for a music player application. The application allows users to play random songs from a predefined set of audio files. The code utilizes the Tkinter and Pygame libraries for creating the graphical user interface and handling audio playback, respectively.Also the playlist created has undergone shuffling which is completely random and uniform.

2 Code Explanation

The code consists of several components, including the initialization of required modules, the definition of variables and functions, and the implementation of music playback controls. Here's a breakdown of the code:

2.1 Importing Required Modules

The following modules are imported:

- numpy - Used for generating random numbers.
- playsound - Used for playing audio files.
- tkinter - Used for creating the graphical user interface.
- pygame - Used for handling audio playback.

2.2 Defining Variables and Functions

The code defines the following variables and functions:

- audio - An empty list that stores the paths of audio files.
- Play() - A function that generates random song paths and populates the audio list with unique songs.

2.3 Initializing the MusicPlayer Class

The MusicPlayer class represents the music player application. It initializes the user interface elements using Tkinter and sets up the Pygame mixer for audio playback.

2.4 Implementing Music Playback Controls

The MusicPlayer class includes methods for controlling music playback:

- play_music() - Plays a random song from the audio list using the Pygame mixer.
- next_music() - Stops the current song playback and plays the next random song from the `audio` list.
- pause_music() - Pauses the currently playing song.
- unpause_music() - Resumes the paused song.

2.5 Creating the Application Window

The script creates a Tkinter window and an instance of the MusicPlayer class, passing the window object as a parameter. The Tkinter event loop is started using the root.mainloop() function.

3 Images

1) The image 1 is of the basic GUI created.
2) The image 2 is of terminal and the GUI while songs are being played.

The code for this **Audio Player** can be found at **https://github.com/ksananth4424/AI1110/blob/main/Audio_Player/audioplayer.py**

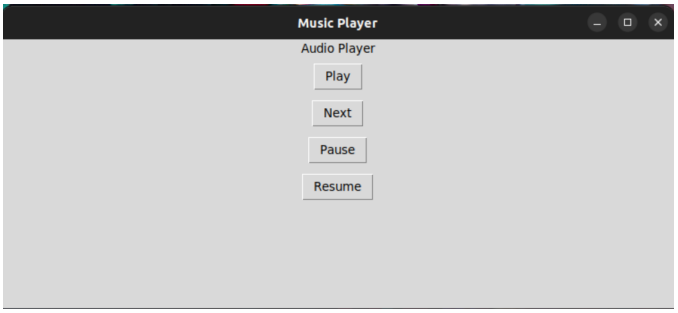

Fig. 1: GUI of Audio Player

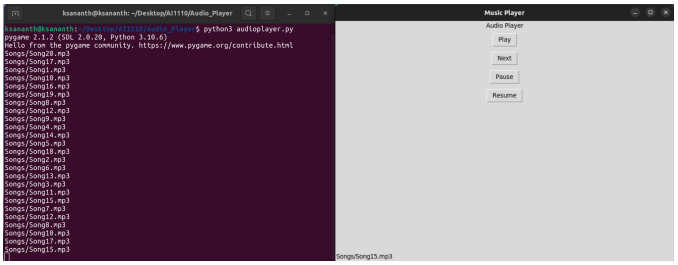

Fig. 2: List of songs that are played

4 Conclusion

The presented code demonstrates a simple music player application with basic playback controls. It utilizes the Tkinter and Pygame libraries to provide a graphical user interface and handle audio playback. By understanding the code's structure and components, you can further customize and enhance the music player functionality.